REMARKS

Claims 1-21 are pending. Claim 6 is amended. Reconsideration of the claims is respectfully requested in view of the following remarks.

I. 35 U.S.C. § 112, Alleged Lack of Enablement for Claim 6

The Office rejects claim 6 under 35 U.S.C. § 112, first paragraph, as failing to comply with the enablement requirement. The Office Action states that it is unclear what a "class line" is, as the specification and drawings do not appear to disclose or enable the term "class line." Applicants submit that a person of ordinary skill in the art would be aware of what a "class line" is; however, to advance prosecution, claim 6 is amended to remove recitation of a "class line."

Therefore, Applicants respectfully request withdrawal of the rejection of claim 6 under 35 U.S.C. § 112, first paragraph.

II. 35 U.S.C. § 102, Alleged Anticipation of Claims 1, 3, 9, 11, and 15-17

The Office rejects claims 1, 3, 9, 11, and 15-17 under 35 U.S.C. § 102(e) as allegedly being anticipated by *Futral et al.* (U.S. Publication No. 2005/0033874). This rejection is respectfully traversed.

Futral teaches direct memory access using memory descriptor list. An operating system may allocate pages to a buffer and may build a memory descriptor list that references the pages allocated to the buffer. A direct memory access (DMA) controller may process the memory descriptor list and transfer data between a buffer defined by the memory descriptor list and another location per the memory descriptor list. See *Futral*, Abstract.

In contradistinction, the present invention, as recited in claims 1, 9, and 15, for example, provides a processor that is configured to generate DMA commands for the management of a cache on the execution of a software program on the processor. The DMA controller is coupled to the processor and is configured to execute the DMA commands for the management of a cache.

With respect to claims 1, 9, and 15, the Office Action alleges that *Futral* teaches a processor that generates DMA commands for the management of a cache, and that the DMA controller is configured to execute the DMA commands for the management of a cache at paragraphs [0001], [0011], and [0015]. The cited portions of *Futral* are as follows:

> [0001] In a computing device, a processor may offload a data transfer to a direct memory access (DMA) engine or controller. In response to a data transfer request, the processor may execute a device driver. The device driver may cause the processor to generate one or more DMA descriptors defining the data transfer. The DMA controller may then process the DMA descriptors and transfer data per the DMA descriptors.
>
> [0011] Embodiments of the invention may be implemented in hardware, firmware, software, or any combination thereof. Embodiments of the invention may also be implemented as instructions stored on a machine-readable medium, which may be read and executed by one or more processors. A machine-readable medium may include any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computing device). For example, a machine-readable medium may include read only memory (ROM); random access memory (RAM); magnetic disk storage media; optical storage media; flash memory devices; electrical, optical, acoustical or other forms of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.), and others. Further, firmware, software, routines, and/or instructions may be described herein as performing certain actions. However, it should be appreciated that such descriptions are merely for convenience and that such actions in fact result from computing devices, processors, controllers, or other devices executing the firmware, software, routines, instructions, etc.
>
> [0015] The chipset 112 may also comprise a direct memory access (DMA) engine or controller 122. The DMA controller 122 may process a DMA command structure 124 and perform data transfers that involve buffers 126 and/or I/O devices 120 per the DMA command structure 124. In one embodiment, the buffers 126 may be virtually contiguous but physically non-contiguous. It should be appreciated that virtually contiguous buffers 126 may permit software and firmware modules such as, for

> example, application 108 to easily manipulate the buffer 126 via a range of sequential virtual addresses. Further, not requiring the buffer 126 to be physically contiguous may enable the operating system 104 to more effectively manage the memory 116. However, the DMA controller 122 may be limited to accessing memory 116 via physical addresses. Accordingly, the DMA command structure 124 directs the DMA controller 122 to the potentially physically non-contiguous buffer fragments 128 that comprise the buffers 126.

The cited portions of *Futral* fail to teach or suggest DMA commands for the management of a cache. In fact, the word "cache" does not appear anywhere in the *Futral* reference. The Office Action is silent as to how the cited portions somehow teach or suggest a DMA controller that is configured to execute DMA commands for the management of a cache. *Futral* fails to teach a cache of any kind.

The Office Action alleges that *Futral* teaches a cache at reference number 16 in FIG. 1. The cited portion is as follows:

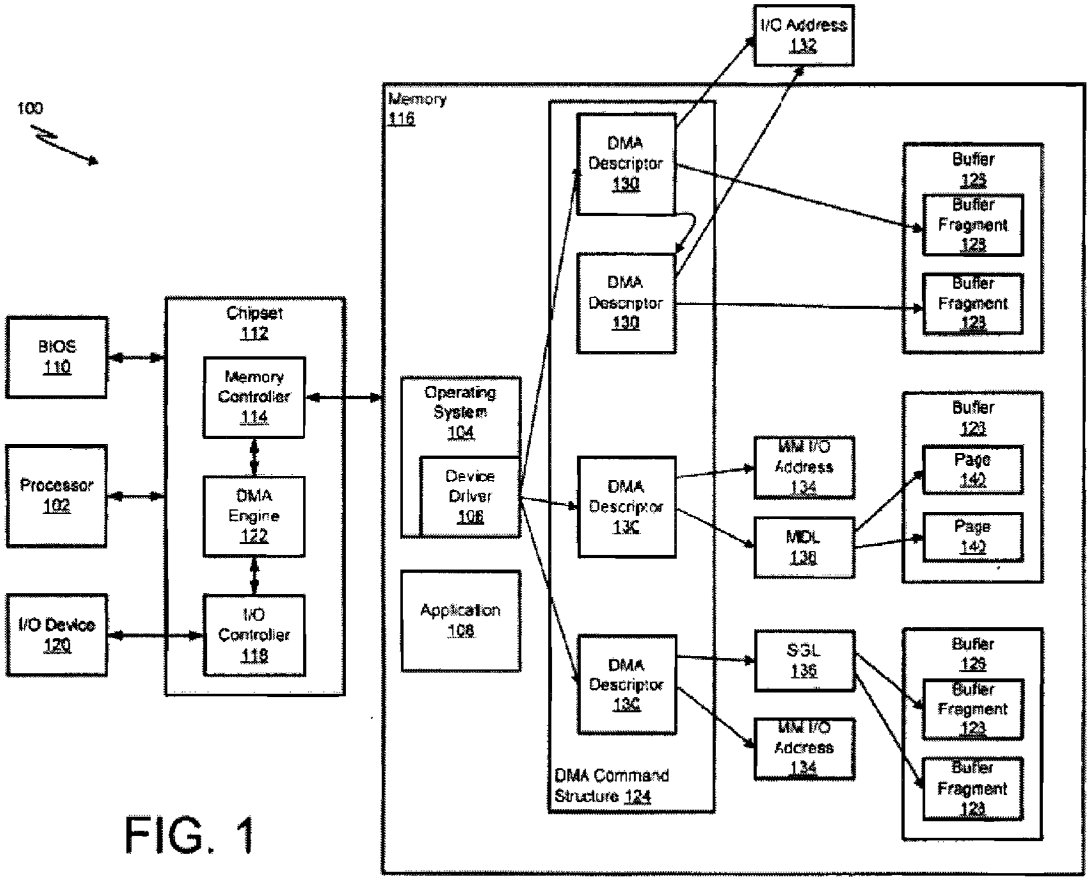

FIG. 1

As seen in the figure, reference number 116 represents a memory; however, there is no teaching in *Futral* of a **cache**. Therefore, the applied reference cannot teach or fairly suggest the further limitation of a DMA controller that is configured to execute DMA commands for the management of a cache.

The applied reference fails to teach or suggest each and every claim limitation; therefore, *Futral* does not anticipate claims 1, 9, and 15. Since claims 3, 11, 16, and 17 depend from claims 1, 9, and 15, the same distinctions between *Futral* and the invention recited in claims 1, 9, and 15 apply for these claims. In addition, claims 3, 11, 16, and 17 recite further combinations of features not taught or suggested by the prior art.

Therefore, Applicants respectfully request withdrawal of the rejection of claims 1, 3, 9, 11, and 15-17 under 35 U.S.C. § 102.

III. 35 U.S.C. § 103, Alleged Obviousness of Claims 2, 4, 5, 7, 8, 10, 12-14, 18, 19, 20, and 21

The Office rejects claims 2, 7, 8, 10, 13, 14, 20, and 21 under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Futral* in view of *Olivier et al.* (U.S. Patent No. 6,738,881). This rejection is respectfully traversed.

With respect to claims 2, 7, 8, 10, 13, 14, 20, and 21, the Office Action acknowledges that *Futral* does not teach a cache coupled to the DMA controller where the system is configured for the execution of the DMA commands for the management of a cache on the DMA controller to manage the operation of the cache coupled to the DMA controller. The Office Action alleges that *Olivier* teaches a plurality of first-in-first-out (FIFO) buffers and where the FIFOs are utilized to manage the data transferring to and from the plurality of memories.

Applicants submit that, like *Futral*, *Olivier* fails to teach or suggest a cache or a DMA controller configured to execute DMA commands for cache management. While FIFO buffers may be used to transfer data, these buffers require no management. A buffer is not a cache. A cache is a storage that keeps frequently accessed data or program instructions readily available so the device, in this case a DMA controller, does not access them repeatedly from slower storage. Because of the nature of a cache, a device requires cache management to maintain the integrity of the data (or instructions) in the cache. On the other hand, a buffer is merely a one-time staging area for data being transferred. By its very nature, a FIFO buffer requires no management. That is, the first data in is the first data out. There is no cache coherency or data integrity problem with a FIFO buffer. Therefore, the system of *Olivier* has no need for special DMA commands for cache management.

Neither *Futral* nor *Olivier* teaches or suggests a cache or a DMA controller that is configured to execute DMA commands for cache management. Even assuming, *arguendo*, that a person of ordinary skill in the art would have found it obvious to combine *Futral* and *Olivier*, the combination would not result in the invention recited in claims 2, 7, 8, 10, 13, 14, 20, and 21. Therefore, *Futral* and *Olivier* do not render claims 2, 7, 8, 10, 13, 14, 20, and 21 obvious. Applicants respectfully request withdrawal of the rejection of claims 2, 7, 8, 10, 13, 14, 20, and 21 under 35 U.S.C. § 103.

The Office rejects claims 4, 5, 12, 18, and 19 under 35 U.S.C. § 103(a) as allegedly being unpatentable over *Futral* in view of *Liao et al.* (U.S. Patent No. 6,681,296). This rejection is respectfully traversed.

With respect to claims 4, 5, 12, 18, and 19, the Office Action acknowledges that *Futral* does not teach or suggest a DMA command that is a flush command or a zero command. The Office Action alleges that *Liao* teaches DMA commands that include a "block flush" command or a "block set to zero" command at col. 3, lines 8-23, which states:

> One strategy that has been used in the past in connection with caches to improve application performance is to provide in the instruction set of the microprocessor a mechanism that enables software assisted cache management. Most modern microprocessors provide instructions in the instruction set which enable software to assist the cache management hardware to some degree in managing the cache. For example, the PowerPC architecture contains several user-accessible instructions in the instruction set for manipulating the data cache that can significantly improve overall application performance. These instructions are: "block touch" (dcbt); "block touch for store" (dcbtst); "block flush" (dcbf); "block store" (dcbst); and "block set to zero" (dcbz). see Zen and the Art of Cache Maintenance, Byte Magazine, March 1997.

The above cited portion explicitly teaches a mechanism in the instruction set **of the microprocessor** that enables software assisted cache management. However, the *Liao* reference fails to teach or suggest providing **DMA commands** for cache management or a DMA controller that is configured to execute DMA commands for cache management. *Futral* also fails to teach or suggest a DMA controller that is configured to execute DMA commands for cache management. Even assuming, *arguendo*, that a person of ordinary skill in the art would have found it obvious to combine *Futral* and *Liao*, the combination would not result in the invention recited in claims 4, 5, 12, 18, and 19. Therefore, *Futral* and *Olivier* do not render claims 4, 5, 12, 18, and 19 obvious. Applicants respectfully request withdrawal of the rejection of claims 4, 5, 12, 18, and 19 under 35 U.S.C. § 103.

IV. Conclusion

It is respectfully urged that the subject application is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Respectfully submitted,

DATE: December 22, 2006

Stephen J Walder Jr

Stephen J. Walder, Jr.
Reg. No. 41,534
ATTORNEY FOR APPLICANTS

Stephen R. Tkacs
Reg. No. 46,430
WALDER INTELLECTUAL PROPERTY LAW, P.C.
P.O. Box 832745
Richardson, TX 75083
(214) 722-6422
AGENT FOR APPLICANTS